

# Improved Draws for Highland Dance

Tim B. Swartz \*

## Abstract

In the sport of Highland Dance, Championships are often contested where the order of dance is randomized in each of the four dances. As it is a widely held belief that competing in the latter sets is advantageous, this paper develops algorithms where balance is achieved in the dance order. Specifically, draws are randomly generated where the sum of the dance orders over all four dances is constant for each dancer.

**Keywords :** Highland Dance, magic rectangles, random number generation, restricted permutations.

---

\*Tim Swartz is Professor, Department of Statistics and Actuarial Science, Simon Fraser University, 8888 University Drive, Burnaby BC, Canada V5A1S6. The author has been partially supported by the Natural Sciences and Engineering Research Council of Canada. The author thanks Mary Munro (SOBHD judge) and Philippa Swartz (dancer) for helpful discussions concerning Highland Dance.

# 1 INTRODUCTION

The origin of Highland Dance precedes recorded history and has today evolved into a highly technical and athletic sport (Flett and Flett 1964). It might be said that the modern period of competitive Highland Dance began in 1950 when the Scottish Official Board of Highland Dancing (SOBHD) was formed. The SOBHD is the umbrella organization for Highland Dancing Associations worldwide and is responsible for many aspects of Highland Dance including the standardization of dance steps, the rules of competition and the certification of instructors.

The most prestigious competitions sanctioned by the SOBHD are known as Championships where the competitors dance four dances; the Fling, the Sword, the Sean Truibhas and the Reel. In a Championship, typically four competitors simultaneously dance the Fling, and are judged. This is referred to as the first set. Then in the second set, four more competitors perform and are judged not only with respect to one another, but also in comparison to those who danced in the first set. The sets continue until all competitors have completed the Fling. The same process is then followed for the Sword, the Sean Truibhas and the Reel. Championship placings are given out based on the overall performance in all four dances. The website [www.toeandheel.com](http://www.toeandheel.com) is a good resource for all matters related to Highland Dance.

Although the SOBHD has many rules, the SOBHD does not dictate the order in which competitors dance the four dances. Some Championships adhere to the rule whereby competitors dance in the inverse order in which their entry form was received. However, many Championships do not proceed according to the “first to register, last to dance” rule. It is a widely held belief that competing in the latter sets is advantageous. Therefore many Championships instead hold a draw where the dance order is randomized for each of the four dances. Random draws are easily implemented, perhaps by choosing numbers

from a hat without replacement.

An example of a draw is given in Table 1 which refers to the 10-11 year old age group in the ScotDance Canada Championship Series 2006 - Canadian Inter-Provincial Championship, July 3, 2006. The ScotDance Canada Championship Series (SDCCS) is an annual event and is the largest Highland Dance gathering in North America. The SDCCS typically attracts in the neighbourhood of 1000 dancers. To interpret Table 1, we observe that there are  $n = 22$  competitors where the third competitor danced in the 18-th position in the Fling, the 7-th position in the Sword, the 11-th position in the Sean Truibhas and the 12-th position in the Reel. Although the draw was “fair” in the sense that each dance was based on a random permutation of the integers  $1, \dots, n$ , it is clear that some competitors danced on average much earlier/later than others. For example, the 14-th competitor danced in the early sets in each of the four dances.

The motivation of this paper is the development of algorithms which produce random draws that are more balanced. Specifically, we attempt to generate draws where the sum of the dance orders over all four dances is constant for each competitor. Introducing some notation, we refer to  $X = (x_{ij}) = (x_1, x_2, x_3, x_4)$  as a draw matrix where  $x_{ij}$  denotes the dance order of competitor  $i = 1, \dots, n$  in dance  $j = 1, 2, 3, 4$ . Then the columns  $x_1, x_2, x_3, x_4$  each consist of a permutation of the integers  $1, \dots, n$ . Our balance requirement imposes the added restriction

$$x_{i1} + x_{i2} + x_{i3} + x_{i4} = 2(n + 1), \quad i = 1, \dots, n. \quad (1)$$

Restricted draws in Highland Dance are related to magic rectangles in combinatorics (Hagedorn 1999). A normal magic rectangle is an  $n$  by  $m$  matrix consisting of the integers  $1, \dots, nm$  where every row sums to the row constant  $m$  and every column sums to the column constant  $n$ . The existence and construction of magic rectangles and their variants

Table 1: Actual draw for the 10-11 year age group at the SDCCS 2006 Canadian Inter-Provincial Championship.

Competitor	Fling	Sword	ST	Reel
1	14	5	13	1
2	5	6	5	11
3	18	7	11	12
4	21	13	6	22
5	11	20	12	20
6	15	15	2	4
7	20	9	7	2
8	3	8	15	7
9	17	12	14	16
10	16	17	1	8
11	1	10	20	19
12	22	21	4	14
13	10	19	19	5
14	4	3	3	6
15	2	22	16	15
16	8	1	18	3
17	12	16	22	9
18	7	18	8	13
19	6	2	10	21
20	13	11	9	18
21	19	4	21	10
22	9	14	17	17

are two questions that are of interest to researchers. Restricted draws in Highland Dance are also related to various experimental designs that arise in statistics. For example, the search and enumeration of orthogonal arrays are important in achieving balance in fractional factorial designs. Design choice is also important with respect to the confounding of variables. Li, Deng and Tang (2004) use a minimum G-aberration criterion in obtaining good designs.

In section 2, we begin by exploring the vast space of draw matrices which satisfy the balance constraint (1). This leads to an algorithm for generating draws that does not require the enumeration of the space. A downside of the algorithm is that it can be computationally demanding for large values of  $n$ . To this end, we present a simpler and faster algorithm that generates from a subspace of the original space. In section 3, we discuss the implementation of the algorithm and its potential use in the competitive world of Highland Dance.

## 2 GENERATING RESTRICTED PERMUTATIONS

When generating uniformly from a finite set, one of the first approaches that comes to mind is to enumerate the set, put the elements of the set in a 1:1 correspondence with the integers  $1, \dots, N$ , and then invoke a uniform generator on  $\{1, \dots, N\}$ . Although simple in theory, this approach is difficult when the enumeration process is challenging.

In the case of generating uniform draw matrices subject to the constraint (1), the enumeration process is challenging. We take an approach where complete enumeration of the set is not necessary. Instead, we generate uniform draw matrices by breaking the process into manageable steps. The first step is the generation of the draw for the Fling. This corresponds to generating the column  $x_1$ , a permutation of the integers  $1, \dots, n$ . The first step is computationally simple and fast, as we implement the pseudo-code:

- let  $x_{i1} = i \quad i = 1, \dots, n$
- for  $m = n, \dots, 2$ 
  - uniformly generate  $k \in (1, \dots, m)$
  - exchange  $x_{m1}$  and  $x_{k1}$

The second step is also computationally simple and fast, as similarly, we generate the column  $x_2$ , a permutation of the integers  $1, \dots, n$  for the Sword.

At this point, it is instructive to reflect on the scope of the remaining problem; the generation of  $x_3$  and  $x_4$  subject to constraint (1). Given  $x_1$  and  $x_2$ , it is possible that there are very few candidates  $x_3$  and  $x_4$  that satisfy (1). For example, if  $x_{i1} = x_{i2} = i$  for  $i = 1, \dots, n$ , then the only solution is  $x_{i3} = x_{i4} = n + 1 - i$  for  $i = 1, \dots, n$ . Alternatively, given  $x_1$  and  $x_2$ , it is possible that there are many candidates  $x_3$  and  $x_4$  that satisfy (1). For example, if  $x_{i1} = i$  and  $x_{i2} = n + 1 - i$  for  $i = 1, \dots, n$ , then  $x_3$  can be any of the  $n!$  permutations of the integers  $1, \dots, n$  and  $x_{i4} = n + 1 - x_{i3}$ ,  $i = 1, \dots, n$ . It is also obvious that given any  $x_1$  and  $x_2$ , there exists at least one solution  $x_3$  and  $x_4$  that satisfies (1).

The third and final step of the algorithm maintains the philosophy of breaking the process into manageable steps. For this, we generate  $x_3$  componentwise, going down (and up) the column vector  $x_3$ . Note that when  $x_{i3}$  is generated,  $x_{i4}$  is determined via  $x_{i4} = 2(n + 1) - x_{i1} - x_{i2} - x_{i3}$ . We define  $I_i$  as the index set for  $x_{i3}$ ,  $i = 1, \dots, n - 1$ .  $I_i$  is the set of candidate values of  $x_{i3}$  given  $x_1$  and  $x_2$ , and given the previously generated values  $x_{13}, \dots, x_{i-1,3}$ . It is straightforward to determine  $I_i$  at any point in the third step. For example,  $I_1 = \{\max\{1, n - x_{11} - x_{12} + 2\}, \dots, \min\{n, 2n - x_{11} - x_{12} + 1\}\}$ . The third step is described by the following pseudo-code:

- $i = 0$
- $i = i + 1$  ★
- if  $i = n$ 
  - then stop
- if  $I_i$  is the empty set
  - then remove  $x_{i-1,3}$  from  $I_{i-1}$  and set  $i = i - 2$
  - else uniformly generate  $x_{i3} \in I_i$
- goto step ★

Since we would like to make the algorithm as efficient as possible, it is clear that the speed of the third step is affected by the frequency with which empty sets are encountered. The occurrence of an empty set implies that the draw matrix with its current values does not satisfy constraint (1). When an empty set is encountered, the algorithm moves one position up the column of  $x_3$  and attempts to generate an acceptable value. Going up the column  $x_3$  corresponds to a rejection step using the rejection algorithm for variate generation (see Evans and Swartz 2000). We have found that a temporary reordering of the dancers is effective in improving the speed of the algorithm. The idea is that we attempt to deal earlier with dancers who have smaller (i.e. more restrictive) index sets. More specifically, we temporarily sort the rows of the draw matrix according to the criterion  $C_i = \min\{x_{i1} + x_{i2}, 2(n + 1) - x_{i1} + x_{i2}\}$  with the top rows of the matrix corresponding to small values of  $C_i$ .

Apart from the simplicity of the proposed algorithm, one of the nice features is that it generates uniform draw matrices from the full space of draw matrices subject to restriction (1). However, when the number of competitors  $n$  is large, the algorithm is sometimes very slow in generating a draw matrix. To this end, we propose a second algorithm which is always fast but generates draw matrices from a subspace of the original space. Fortunately, the subspace is still very large.

The second algorithm proceeds as in the first algorithm by efficiently generating permutations  $x_1$  and  $x_2$ . That is, random draws are generated for both the Fling and the Sword. We then set  $x_{i3} = n + 1 - x_{i1}$  and  $x_{i4} = n + 1 - x_{i2}$  for  $i = 1, \dots, n$ , and this defines a draw matrix that satisfies (1). We then consider the distinct values of  $x_{i3} + x_{i4}$  for all  $i = 1, \dots, n$ . For each distinct value, we permute the rows of the draw matrix  $X$  which share the value. Clearly, the row permutations leave the restriction (1) intact. Finally, we permute the four columns, and the column permutation also leaves the restriction (1) intact. An example of a draw matrix which satisfies (1) but cannot be generated by the second algorithm is

$$X = \begin{pmatrix} 1 & 1 & 4 & 4 \\ 3 & 3 & 1 & 3 \\ 2 & 4 & 2 & 2 \\ 4 & 2 & 3 & 1 \end{pmatrix}.$$

To get a sense of the relative sizes of the sets of draw matrices, consider the case  $n = 4$ . There are a total of  $(4!)^4 = 331776$  draw matrices obtained by permuting the entries within each of the four columns. However, there are only 2520 draw matrices that satisfy (1), and there are 1944 draw matrices that satisfy (1) and can be generated via the second algorithm.

### 3 IMPLEMENTATION AND DISCUSSION

In this paper, we have proposed two algorithms for the generation of draw matrices in Highland Dancing Championships. The motivation of this work is an attempt to improve draws so that the order in which competitors dance is balanced across the four dances.



We have implemented the algorithms using a perl script which invokes a Fortran programme. The software is free to use and is available at the website

[www.stat.sfu.ca/~tim/highland/highland.html](http://www.stat.sfu.ca/~tim/highland/highland.html)

The interface is simple. A user is required to submit the number of dancers  $1 \leq n \leq 99$  and an integer seed. The purpose of the seed is to allow officials to replicate draws when required and to generate different random draw matrices.

The perl script has been coded such that the first algorithm is invoked when  $n \leq 28$ , and the second algorithm is invoked when  $n > 28$ . In our investigations, when  $n = 28$ , a user should not have to wait more than a minute to obtain a generated draw. Note that when  $n$  is large, say  $n = 70$ , the first algorithm may require several days of computation with some seeds. Our choice of  $n = 28$  as the cutoff point was based on computational speed and was also based on a practical matter. In the annual Canadian Inter-Provincial Championship, there are typically at most 28 competitors. These competitors consist of the three representatives from each of 9 provinces and the former Canadian Champion. For illustration of our software, in Table 2, we provide a hypothetical draw for the 10-11 year old age group at the SDCCS 2006 Canadian Inter-Provincial Championship based on the integer seed 4327.

We see the possibility of the generalization of the ideas found in this paper to other sports where aggregate prizes are awarded based on the performance in  $p$  sub-events. In this case, a random draw with  $n$  competitors would correspond to an  $n$  by  $p$  matrix rather than an  $n$  by 4 matrix. Events such as the men's individual all-around competition in gymnastics comes to mind where this is based on  $p = 6$  sub-events.

Finally, we mention that our software is being presented to the executive members of ScotDance BC and ScotDance Canada for consideration in future Championships.

Table 2: Hypothetical draw for the 10-11 year age group at the SDCCS 2006 Canadian Inter-Provincial Championship.

Competitor	Fling	Sword	ST	Reel
1	18	15	2	11
2	7	16	22	1
3	2	11	12	21
4	13	5	21	7
5	19	10	11	6
6	4	21	19	2
7	21	6	5	14
8	15	2	10	19
9	9	19	6	12
10	12	1	20	13
11	11	8	9	18
12	8	9	13	16
13	16	22	4	4
14	1	12	18	15
15	17	18	1	10
16	14	20	3	9
17	5	14	7	20
18	20	13	8	5
19	3	4	17	22
20	22	7	14	3
21	10	3	16	17
22	6	17	15	8

## 4 REFERENCES

- Evans, M. and Swartz, T.B. (2000). *Approximating Integrals via Monte Carlo and Deterministic Methods*, Oxford University Press.
- Flett, J.F. and Flett, T.M. (1964). *Traditional Dancing in Scotland*, London: Routledge & Paul.
- Hagedorn, T.R. (1999). “Magic rectangles revisited”, *Discrete Mathematics*, (207), 65-72.
- Li, Y., Deng, L.Y. and Tang, B. (2004). “Design catalog based on minimum G-aberration”, *Journal of Statistical Planning and Inference*, (124), 219-230.